

CLUBE

Z

80

/83

N.º 9

CULTURE



80/

83/

N.º 3

CLUBE Z-80
NUMERO 9 / 30 JUNHO 1983
AV. BOAVISTA, 832 - 2 T. Telef. 65127
4100 PORTO

N E S T E N U M E R O

Programacao Estruturada.....2

PROGRAMAS ZX 81

6 programas para iniciar as criancas.....4

Graficos ZX 81.....8

PROGRAMAS SPECTRUM

Puck Man.....12

Aliens.....15

O som do Spectrum.....17

Climas-Graficos ombrotermicos.....18

PROGRAMAS NEW-BRAIN

Metodo de previsao.....19

Livros Spectrum: breve panoramica.....22

Introducao a Linguagem Maquina (cont).....23

PROGRAMACAO ESTRUTURADA

in. "STRUCTURED COBOL". PHILIPPAKIS Andreas e KAZMIER Leonard

Traduzido e adaptado ao BASIC por
FRANCISCO J.R. SANTOS

Espero que este trabalho tenha utilidade para os socios do Clube Z-80, para o qual o adaptei e compus, a partir de uma traducao que fiz ha cerca de dois anos do livro STRUCTURED COBOL de Andreas S. Philippakis e de Leonard J. Kazmier, ambos professores da Arizona State University.

Apesar de este artigo nao ser originalmente escrito para a linguagem BASIC, mas sim para o COBOL, acho que tem bastante utilidade para o BASIC, uma vez que os principios de programacao estruturada sao os mesmos.

Como tal fiz uma adaptacao para o BASIC, aproveitando o original. Dai que haja a certa altura uma parte em que tive de por duas NOTAS, para que assim quem quer que fosse ler este trabalho, e que nao percebesse nada de COBOL, pudesse entender melhor.

Nao quis tambem retirar duas instrucoes que embora sejam de COBOL, tambem se aplicam nalguns micros: WHILE e UNTIL. A instrucao IF-THEN-ELSE, ja e mais aplicada, assim como a instrucao ON X GO TO A, B, C, que ja pertencem ao BASIC que se pode considerar de uma forma STANDARD.

INTRODUCAO

O modo tradicional em que tem sido vista a programacao, e como uma criacao por um individuo. As caracteristicas de um bom programador tem sido a habilidade de escrever "inteligentes" programas, "inteligentes" que por vezes tem sido sinonimo de complexo e obscuro. O problema e que a codificacao de programas "obscuros" e dificil de compreender por outras pessoas para alem do autor, e mesmo esse pode ter dificuldades sobre o mesmo, meses ou anos mais tarde.

A mudanca de pessoal e a mudanca por necessidade do negocio tem forçado a atencao na necessidade de escrever programas correctos e facis de manter. Esta necessidade tem

sido reconhecida, e a ela tem sido dado destaque nos ultimos anos.

O termo de programacao estruturada tem vindo a significar a colecao de pensamentos, principios e praticas que resultou dos esforcos de escrever programas correctos e facis de manter.

Como com todas as novas materias, ha um consideravel desacordo relativamente a como e e como nao e a programacao estruturada. Nao ha uma definicao concreta de programacao estruturada neste texto; Isso sera evitado, ainda que qualquer tentativa seja provavel, breve, util e muito limitada.

ESTRUTURA DO PROGRAMA

Um programa de computador é um conjunto de instruções para um computador. (...) Linguagens de alto-nível, como o BASIC, são concebidas para o uso humano na sua forma directa e são concebidas para a máquina, embora indirectamente, através da compilação. Portanto, o básico princípio de uma boa programação é que o programa deve ser compreensível.

Um programa pode ser compreendido, pode ser analisado e testado para verificar a sua correcção. Programas incorrectos têm dado problemas desde os primeiros tempos (...).

Outro princípio básico da boa programação é escrever programas correctos. Num modo mais teórico, falamos da prova para os programas correctos, com esforcadas reflexões para provar em termos lógicos e não ambiguos, que um bocado de programa codificado é correcto, p. ex., que cumpre o objectivo e nada mais. O progresso teórico tem sido feito só num contexto muito limitado, e nesta altura não podemos confiar num número de regras e condutas que se aproximam mais ao senso comum que a forma teórica.

O programa estrutura é um conceito de senso comum. Significa que o programa não deve ser instruturado (...).

Um programa estruturado é caracterizado pela clareza e simplicidade na sua lógica da estrutura de fluxo, lê-se como uma língua vulgar do princípio ao fim, em vez de ramificar por parágrafos desde os primeiros até aos últimos e voltar atrás, quase como uma bola de ping-pong. Por isso, uma simples e directa linha de fluxo lógica é um outro bom princípio de programação.

É claro, tarefas de programação são já por natureza complexas. É claro que podemos levantar a questão óbvia: como podemos evitar a complexidade de

tarefas inerentes?

Os programas estruturados reduzem e controlam a complexidade, mas não a eliminam. Para além disso, os programas estruturados conseguem clareza e simplicidade entre saltos de segmentos ou modelos de um programa. O Homem é caracterizado por uma racionalidade limitada. Então, quando confrontado com tarefas complexas, tenta dividi-las em partes pequenas.

Num modo igual, os programas estruturados consistem em pequenas partes interligadas. Cada parte ou módulo é simples e compreensível por si só, e em cada um se pode encontrar simplicidade e clareza. É necessária uma atenção especial na especificação das ligações entre si.

Preferivelmente, um módulo existe como uma identidade separada que pode ser usada numa variedade de contextos, sem se modificarem, excepto na especificação da ligação para cada uma das vezes em que esse módulo é usado. Deste modo, um módulo pode ser trabalhado independentemente, para qualquer programa do qual faça parte, sem serem feitas iniciais ou subsquentes modificações.

Temos assim discutido outro princípio para uma boa programação: um programa estruturado consiste na interligação de módulos, sendo cada um simples e claro tal como deve ser concebido para o nosso trabalho.

Uma estrutura modular permite a abstracção do resto do trabalho, e proporciona a mente a possibilidade de estar a altura da complexidade desse módulo.

CONTINUA
NO
PROXIMO NUMERO

MICROCOMPUTADOR SINCLAIR TIMEX ZX 81

COMO AJUDAR AS CRIANCAS A USAR O COMPUTADORAPRENDENDO !

Tenho tres criancas, com idades entre os 4 e os onze anos e sempre que posso preocupo-me com a educacao das criancas. Recentemente tive varios encontros com professores primarios e professores das Escolas do Magisterio Primario, e foi interessante observar (e discutir) o interesse pro e contra, manifestado no sentido da espantosa ajuda, que o microcomputador pode dar ao professor e aos pais.

Um dos pontos habitualmente dificeis e repetitivos no ensino, e por exemplo o da contagem e o da aprendizagem das tabelas de calculo e ainda o desenvolvimento do calculo mental.

Uma das questoes que pode espantar os leigos, e precisamente o facto de que o microcomputador pode auxiliar a crianca a desenvolver a sua rapidez de calculo mental, contrariamente a ideia de que a maquina so executa tarefas e portanto a crianca nao necessita de saber calcular!

OS TRES PROGRAMAS INICIAIS SAO MUITO SIMPLES.

Introduzem a crianca (entre os tres e os seis anos) no uso dos simbolos alfanumericos, e no dos comandos RUN e ENTER (NEW LINE).

Habitualmente as criancas ficam fascinadas quando veem os seus nomes (ou os nomes de quem gostam) no ecran do televisor. Podem ainda observar no Programa 4 'LETRAS GRANDES' tres palavras de quatro letras cada. Cada letra grande e composta de varias letras pequenas, do caracter escolhido para exibicao. No caso de pretender alterar para quatro linhas com nove caracteres por linha, podera modificar na linha 270 ... for n=1 to 6 e ainda mudar a linha 280plot x + x1.6-y

AS CRIANCAS ENTRE OS SEIS E OS DEZ ANOS, adoram dois tipos de programas classicos ARITMETICA em que sao ajudados a desenvolver a sua pratica e o jogo das PALAVRAS

O PROGRAMA 5 ira gerar equacoes aritmeticas de dois numeros, e que incluem os sinais (+, -, *, /)entre os dois numeros. A crianca devera entrar com o numero correcto para a resposta. Se a resposta e errada, e convidado a tentar de novo. Apos tres respostas erradas, o nosso amigo ZX dara a resposta certa e arranca com novo caso.

O PROGRAMA 6 permite-lhe entrar com dez palavras de ate dez letras cada. Apos o que a crianca vai dar entrada das letras respectivas e ser eventualmente corrigida. Normalmente estara ocupada durante cerca de meia hora.

QUALQUER UM DESTES PROGRAMAS PODERA FACILMENTE SER ADAPTADO A OUTRA MAQUINA, ESCOLHEMOS O ZX 81 PORQUE E ECONOMICO E QUEM JA POSSUI OUTRA MAQUINA PODERA COM FACILIDADE EMPRESTAR O ZX AS CRIANCAS.

PROGRAMA 1

```

-----
10 CLS
20 LET Z$=INKEY$
30 FOR N=1 TO 21
40 PRINT Z$
50 NEXT N
60 GO TO 10

```

PROGRAMA 2

```

-----
10 INPUT A$
20 FOR N=1 TO 21*31/LEN A$
30 PRINT A$;" ";
40 NEXT N
50 CLS
60 GO TO 10

```

PROGRAMA 3

```

-----
10 INPUT A$
20 SCROLL
30 PRINT A$
40 GO TO 10

```

PROGRAMA 5

```

-----
1 REM ARITMETICA
10 LET A=1+INT(10*RND)
20 LET B=1+INT(10*RND)
30 LET C=21+INT(4*RND)
38 LET X=0
39 SCROLL
40 PRINT A;" ";CHR$ C;" ";B;" ";
   " "; "="; "?"
50 INPUT D
PALAVRAS"

```

```

60 IF C=21 THEN LET E=A+B

```

```

70 IF C=22 THEN LET E=A-B
80 IF C=23 THEN LET E=A*B
90 IF C=24 THEN LET E=A/B
100 IF ABS(D-E) <=.001 THEN GOTO
   1000
110 GOTO 2000
120 GOTO 10
1000 SCROLL
1005 PRINT E
1009 SCROLL

```

PROGRAMA 6

```

-----
1 REM PALAVRAS
100 GOSUB 1000
110 CLS
120 FOR I=1 TO 10
125 CLS
130 GOSUB 2000
140 NEXT I
150 GOTO 100
1000 PRINT "DAR ENTRADA DE 10

```

```

1001 PRINT "COM MENOS DE 11 LETRAS
      CADA"

```

```

1002 PAUSE 120
1005 DIM A$(10,10)
1010 FOR I=1 TO 10
1020 CLS
1030 PRINT I
1040 INPUT A$(I)
1050 NEXT I
1060 RETURN
2000 FOR X=1 TO 10

```


PROGRAMA 5 (continuação)

```

1010 PRINT "EXCELENTE ! "
1015 PRINT "NOVA TENTATIVA"
1020 GOTO 10
2000 LET X=X+1
2010 PRINT "ERRO ! TENTA DE NOVO"

2020 IF X>=3 THEN GOTO 10
2030 GOTO 39

```

PROGRAMA 6 (continuação)

```

2010 IF CODE A$(I,X)= THEN GOTO
2050

2030 PRINT "#";
2040 NEXT X
2050 FOR Y=1 TO 10
2060 PRINT AT Y+2,0;"ESCREVE UMA
LETRA"
2070 INPUT B$
2080 PRINT AT Y+3,0;B$
2085 PAUSE 120
2090 IF LEN B$=X-1 THEN GOSUB 5000
2100 LET Z=0
2110 FOR W=1 TO X
2115 FOR U=1 TO LEN B$
2120 IF CODE B$(U)=CODE A$(I,W)
THEN GOSUB 3000
2130 NEXT W
2135 NEXT U
2140 IF Z=0 THEN GOSUB 4000
2150 NEXT Y
2160 CLS
2170 PRINT "ESTAS <PENDURADO > !"
2180 PAUSE 120
2190 RETURN
3000 PRINT AT 0,W-1;B$(U)
3010 LET Z=Z+1
3020 RETURN
4000 LET V=V+1

4010 IF V=1 THEN PRINT AT V,25;"TU"
4020 IF V=2 THEN PRINT AT V,25;"TIVESTE"
4030 IF V=3 THEN PRINT AT V,25;"MUITAS"
4040 IF V=4 THEN PRINT AT V,22;"RESPOSTAS"
4050 IF V=5 THEN PRINT AT V,25;"ERRADAS"
4080 RETURN
5000 LET M=0
5010 FOR N=1 TO X-1
5020 IF B$(N)=A$(I,N) THEN LET
M=M+1
5030 NEXT N
5040 IF M=X-1 AND I=10 THEN LET
M=M+1
5045 IF M=X-1 THEN NEXT I
5045 RETURN

```


PROGRAMA 4 (ZX 81)

EDUCAÇÃO

```

1 REM :LETRAS GRANDES*
100 GOTO 500
110 LET X1=0
120 FOR I=1 TO LEN P$
130 LET J=CODE P$(I)
150 FOR Y=0 TO 7
160 LET K=PEEK (7680+J*8+Y)
170 LET L=128
180 FOR X=0 TO 7
190 IF K<L THEN GOTO 220
200 PRINT AT Y+15,X+X1;CHR$(J)
210 LET K=K-L
220 LET L=L/2
230 NEXT X
240 NEXT Y
250 LET X1=X1+7
260 NEXT I
270 FOR N=1 TO 8
280 SCROLL
290 NEXT N
300 RETURN
500 INPUT A$
510 INPUT B$
520 INPUT C$
530 INPUT D$
540 LET P$=A$
550 GOSUB 110
560 LET P$=B$
570 GOSUB 110
580 LET P$=C$
590 GOSUB 110
600 LET P$=D$
610 GOSUB 110
620 GOTO 540

```

Relativamente ao gerador de caracteres (questão de Fernando Pereira / Cacia) sugerimos-lhe que experimente o programa 4 (nesta página), com as modificações:

- linha 200 PLOT X+X1,6-Y
- LINHA 270 FOR N=1 TO 6

Observa 4 linhas com 9 caracteres por linha.

Se conseguir bons resultados, transmita-nos ou verifique quais os efeitos que não obtém e quais as modificações que gostaria de observar.

Alfredo Alberto (Sintra) pretende / "elementos sobre um soft que existe em França, sob a designação de Q SAVE, e que permite a transferência de programas para a cassete a 4000 bauds. O ZX MONITOR tem algo a ver com este problema? Qual a sua finalidade?"

O programa ZX monitor possui os seguintes utilitários com mais interesse: VERIFY, MERGE, RENUM, COPY, etc. mas não altera a frequência de gravação. Já tentamos passar um programa Q SAVE e sem sucesso. Ficamos convencidos de que o programa teria de funcionar com qualquer coisa de hardware externo. Devolvemos portanto a pergunta aos outros amigos do CLUBE 280. Se algum já experimentou o Q SAVE, que nos diga da sua experiência.

SOFTWARE

DESCONTOS

OS DESCONTOS QUE FAZEMOS NO PREÇO DO SOFTWARE PARA OS MICROCOMPUTADORES É OBTIDO APENAS QUANDO OS PROGRAMAS SEJAM PEDIDOS NO CLUBE Z-80. EM QUALQUER ENCOMENDAS FEITAS EM FIRMAS COMERCIAIS NÃO PODERÁ HAVER RETORNO DEVIDO A PROBLEMAS FISCAIS QUE ISSO OCASIONARIA.

ZX81

GRAFICOS ZX81

<Adaptado de
"L'Ordinateur
Individuel", 45>

Com este programa para o ZX81 de memoria 16k, pode executar varios graficos, conforme as instrucoes dadas. A entrada de dados pode ser dada em modo de programacao (varias instrucoes seguidas), ou directa (instrucao por instrucao), sendo neste caso necessario, no final de cada execucao, introduzir a instrucao "PROC". Se utilizar "FIM", o programa comecara de novo, para entrada de novos dados. Quando terminado o grafico, aparecera a frase "FIM DE EXECUCAO" e, no canto inferior esquerdo, as opcoes:

"2 - IMPR." - IMPRESSORA,
"N - REGR." - REGRESSAR, para nova entrada de dados.

INSTRUCOES:

AVX - Tracado de uma linha de comprimento X

DRX - Angulo de X graus a direita

GAX - Angulo de X graus a esquerda

CACH - Permite deslocar um ponto no ecran, sem tracar a linha. Anula a instrucao "MONT"

MONT - Instrucao contraria da "CACH"

```

1 REM "ZXLOGO"
2 REM CLUBE Z80
3 DIM F$(1,1)
4 DIM G$(1,1)
5 DIM H$(1,1)
6 DIM I$(1,1)
7 DIM J$(1,1)
8 DIM K$(1,1)
9 DIM L$(1,1)
10 DIM D$(1,1)
11 DIM E$(1,1)
15 GOSUB 4000
19 GOTO 500
20 INPUT A$
21 PRINT AT ZZ,0;A$
22 LET ZZ=ZZ+1
23 IF ZZ>=21 THEN GOSUB 5000
24 IF LEN A$<4 THEN GOTO 30
25 IF A$(1 TO 4)<>"POUR" THEN
GOTO 30
26 IF S<180 THEN GOTO 490
27 PRINT "IMPOSSIVEL: ATINGIU
OS LIMITES"
28 LET U=0
29 RETURN
30 LET C$=""
31 LET Z$=""
32 FOR N=1 TO LEN A$
33 IF CODE A$(N)>37 OR CODE A$
(N)<20 THEN LET Z$=Z$+A$(N)
34 IF CODE A$(N)<=37 AND CODE
A$(N)>20 THEN LET C$=C$+A$(N)
35 NEXT N
36 IF Z$="FIM" THEN LET U=0
37 RETURN
40 IF M$( TO 2)="AV" THEN LET
RM=BM
42 IF M$( TO 2)="GA" THEN LET
A=A+(BM*PI/180)
44 IF M$( TO 2)="DR" THEN LET
A=A-(BM*PI/180)
45 IF M$( TO 2)=">>" THEN LET
A=A+(BM*PI/180)
46 IF M$( TO 2)=">" THEN LET
RM=RM+BM
48 IF M$( TO 2)=">>" OR M$( TO
2)="DR" OR M$( TO 2)="GA" THEN
RETURN
50 IF M$( TO 2)<>"AV" AND M$(1
)<>">" THEN GOSUB 70
51 IF NOT U THEN GOTO 56
52 FOR R=1 TO RM
53 LET XX=X+R*COS A
54 LET YY=Y+R*SIN A
55 IF XX>63 OR YY>43 THEN GOTO
58
56 IF U THEN PLOT XX,YY
58 NEXT R
60 LET X=XX
62 LET Y=YY
66 LET U=1
68 RETURN
70 IF M$<>"CACH" THEN GOTO 86
72 LET U=0
75 LET U=0
80 RETURN
86 IF M$<>"MONT" THEN GOTO 95
87 LET U=0
88 LET U=1
90 RETURN
95 LET TT=220
98 LET TP=6
99 LET T=0
100 LET PA=0
102 IF M$=D$(1) THEN GOTO TT
104 LET T=T+TP
106 LET PA=E
108 IF M$=E$(1) THEN GOTO TT
110 LET T=T+TP
112 LET PA=F
114 IF M$=F$(1) THEN GOTO TT
116 LET T=T+TP
118 LET PA=G
120 IF M$=G$(1) THEN GOTO TT
122 LET T=T+TP
124 LET PA=H
126 IF M$=H$(1) THEN GOTO TT
128 LET T=T+TP

```



```

130 LET PA=I
132 IF M$=I$(1) THEN GOTO TT
134 LET T=T+TP
136 LET PA=J
138 IF M$=J$(1) THEN GOTO TT
140 LET T=T+TP
142 LET PA=K
144 IF M$=K$(1) THEN GOTO TT
146 LET T=T+TP
148 LET PA=L
150 IF M$=L$(1) THEN GOTO TT
152 LET T=T+TP
153 LET Z=1
154 IF M$="PROG" THEN GOTO 156
155 PRINT AT 1,0;"O NOME: """;M
$;" "" " NAO ESTA DEFINIDO"
156 LET P=0
157 LET U=0
158 LET M=0
159 RETURN
160 LET P$=D$(N)
162 LET PP=D(N)
164 RETURN
166 LET P$=E$(N)
168 LET PP=E(N)
128 LET T=T+TP
130 LET PA=I
132 IF M$=I$(1) THEN GOTO TT
134 LET T=T+TP
136 LET PA=J
138 IF M$=J$(1) THEN GOTO TT
140 LET T=T+TP
142 LET PA=K
144 IF M$=K$(1) THEN GOTO TT
146 LET T=T+TP
148 LET PA=L
150 IF M$=L$(1) THEN GOTO TT
152 LET T=T+TP
153 LET Z=1
154 IF M$="PROG" THEN GOTO 156
155 PRINT AT 1,0;"O NOME: """;M
$;" "" " NAO ESTA DEFINIDO"
156 LET P=0
157 LET U=0
158 LET M=0
159 RETURN
160 LET P$=D$(N)
162 LET PP=D(N)
164 RETURN
166 LET P$=E$(N)
168 LET PP=E(N)
170 RETURN
172 LET P$=F$(N)
174 LET PP=F(N)
176 RETURN
178 LET P$=G$(N)
180 LET PP=G(N)
182 RETURN
184 LET P$=H$(N)
186 LET PP=H(N)
188 RETURN
190 LET P$=I$(N)
192 LET PP=I(N)
194 RETURN
196 LET P$=J$(N)
198 LET PP=J(N)
200 RETURN
202 LET P$=K$(N)
204 LET PP=K(N)
206 RETURN
208 LET P$=L$(N)
210 LET PP=L(N)
212 RETURN
220 LET NN=2
223 LET CB=0
224 FOR C=1 TO EE
226 FOR N=NN TO PA
228 GOSUB 160+T
229 IF NOT PP AND P$(<)"CACHE" A
ND P$(<)"MONT" THEN GOSUB 6000
230 IF P$(<)"REPE" THEN GOTO 236
232 LET EE=PP
233 LET N=N+1
234 LET NN=N
236 GOTO 224
238 LET M$=P$

```

```

240 LET BM=PP
242 GOSUB 40
243 NEXT N
244 NEXT C
245 IF CT THEN GOSUB 6100
246 IF CB THEN GOTO 223
247 LET T=0
248 LET EE=1
250 LET U=0
252 RETURN
300 LET D=0
302 DIM D$(D,4)
304 LET D$(1)=A$(6 TO )
306 DIM D(D)
308 FOR T=2 TO 0
310 GOSUB 20
312 LET D$(T)=Z$
314 IF CODE C$ THEN LET D(T)=VA
L C$
316 NEXT T
318 RETURN
320 LET E=0
322 DIM E$(E,4)
324 LET E$(1)=A$(6 TO )
326 DIM E(E)
328 FOR T=2 TO 0
330 GOSUB 20
332 LET E$(T)=Z$
334 IF CODE C$ THEN LET E(T)=VA
L C$
336 NEXT T
338 RETURN
340 LET F=0
342 DIM F$(F,4)
344 LET F$(1)=A$(6 TO )
346 DIM F(F)
348 FOR T=2 TO 0
350 GOSUB 20
352 LET F$(T)=Z$
354 IF CODE C$ THEN LET F(T)=VA
L C$
356 NEXT T
358 RETURN
360 LET G=0
362 DIM G$(G,4)
364 LET G$(1)=A$(6 TO )
366 DIM G(G)
368 FOR T=2 TO 0
370 GOSUB 20
372 LET G$(T)=Z$
374 IF CODE C$ THEN LET G(T)=VA
L C$
376 NEXT T
378 RETURN
380 LET H=0
382 DIM H$(H,4)
384 LET H$(1)=A$(6 TO )
386 DIM H(H)
388 FOR T=2 TO 0
390 GOSUB 20
392 LET H$(T)=Z$
394 IF CODE C$ THEN LET H(T)=VA
L C$
396 NEXT T
398 RETURN
400 LET I=0
402 DIM I$(I,4)
404 LET I$(1)=A$(6 TO )
406 DIM I(I)
408 FOR T=2 TO 0
410 GOSUB 20
412 LET I$(T)=Z$
414 IF CODE C$ THEN LET I(T)=VA
L C$
416 NEXT T
418 RETURN
420 LET J=0
422 DIM J$(J,4)
424 LET J$(1)=A$(6 TO )
426 DIM J(J)
428 FOR T=2 TO 0
430 GOSUB 20
432 LET J$(T)=Z$
434 IF CODE C$ THEN LET J(T)=VA
L C$
436 NEXT T

```



```

438 RETURN
440 LET K=0
442 DIM K$(K,4)
444 LET K$(1)=A$(6 TO )
446 DIM K(K)
448 FOR T=2 TO 0
450 GOSUB 20
452 LET K$(T)=Z$
454 IF CODE C$ THEN LET K(T)=VA
  C$
456 NEXT T
458 RETURN
460 LET L=0
462 DIM L$(L,4)
464 LET L$(1)=A$(6 TO )
466 DIM L(L)
468 FOR T=2 TO 0
470 GOSUB 20
472 LET L$(T)=Z$
474 IF CODE C$ THEN LET L(T)=VA
  C$
476 NEXT T
478 RETURN
490 GOSUB 300+5
492 LET S=S+20
494 LET U=0
496 RETURN
500 CLS
501 LET X=30
502 LET Y=20
504 LET A=0
506 LET EE=1
508 LET Z=EE
510 LET U=Z
512 LET O=Z
514 LET U=Z
516 LET T=0
518 LET U=Z
519 LET ZZ=3
520 LET CT=A
522 LET CB=A
523 LET P$=""
524 LET ST=-1
1000 GOSUB 3000
1020 DIM B$(0,4)
1022 DIM B(0)
1030 FOR M=1 TO 0
1032 GOSUB 20
1034 IF NOT U THEN GOTO 500
1046 LET B$(M)=Z$
1050 IF CODE C$ THEN LET B(M)=VA
  C$
1054 NEXT M
1055 LET MM=1
1056 FOR P=1 TO 0
1057 FOR M=MM TO 0
1058 IF B$(M) <> "REPE" THEN GOTO
1063
1059 LET O=B(M)
1060 LET MM=M+1
1061 LET M=M+1
1062 GOTO 1056
1063 LET M$=B$(M)
1064 LET BM=B(M)
1065 GOSUB 40
1066 NEXT M
1068 NEXT P
1070 IF NOT Z THEN GOTO 1030
1072 PRINT AT 0,0;" "
  EXECUCAO " " AT 20,0;"Z-
  IMPR." AT 21,0;"N-REGR."
1074 LET O=1
1075 PAUSE 4E4
1076 IF INKEY$="Z" THEN COPY
1077 IF INKEY$="N" THEN GOTO 200
1
1092 GOTO 1076
2000 SAVE "ZXLOG"
2001 RUN
3000 PRINT AT 0,0;"DESEJA DAR MA
IS QUE UMA INSTRUCAO (S/
N) ?"
3005 LET U=1
3010 INPUT A$
3015 IF A$="S" THEN LET A$="O"
3016 IF A$(1)="O" THEN GOTO 3100
3020 LET O=1
3025 LET Z=0

```

```

3040 GOTO 3110
3100 PRINT "QUANTAS ?"
3105 INPUT O
3110 PRINT AT 0,0;"

```

```
3111 PRINT "
```

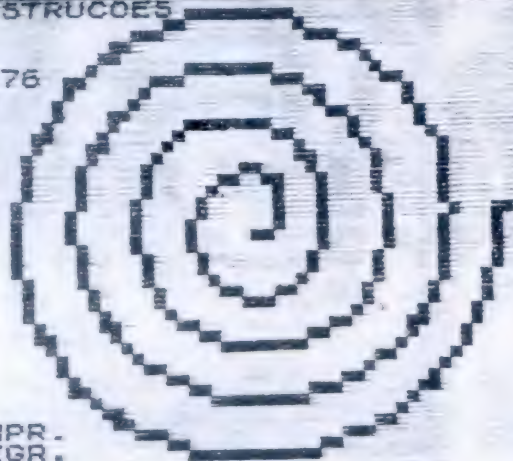
```

3112 PRINT AT 0,0;"DAR ENTRADA O
UANDO APARECER"
3115 IF O <> 1 THEN PRINT AT 1,0;0
  " INSTRUCCOES"
3120 RETURN
4000 LET S=0
4001 LET D=5
4002 LET E=5
4003 LET F=5
4004 LET G=5
4005 LET H=5
4006 LET I=5
4007 LET J=5
4008 LET K=5
4009 LET L=5
4011 RETURN
5000 FOR N=2 TO 21
5005 PRINT AT N,0;"
5010 NEXT N
5015 LET ZZ=2
5020 RETURN
5000 LET SC=C
5005 LET SEE=EE
5010 LET SN=N+1
5015 LET SNN=SNN
5020 LET SPA=PA
5022 LET ST=T
5025 LET CT=1
5030 RETURN
5100 LET C=SC
5105 LET EE=SEE
5110 LET N=SN
5115 LET NN=SNN
5120 LET PA=SPA
5125 LET T=ST
5130 LET CT=0
5135 LET CB=1
5140 RETURN

```

FIN DE EXECUCAO
4 INSTRUCCOES

AV1
REPE76
>>20
>.1



Z-IMPR.
N-REGR.

4 INSTRUÇÕES

AV1
REPE31
DR90
→1



Z-IMPR.
N-REGR.

5 INSTRUÇÕES

CACH
GA90
AV10
DR90
MONT
PEPE36
J2
DR10



Z-IMPR.
N-REGR.

6 INSTRUÇÕES

CACH
GA90
AV10
DR150
MONT
REPE3
AV20
DR120



Z-IMPR.
N-REGR.

9 INSTRUÇÕES

CACH
AV10
GA90
AV10
GA90
MONT
REPE4
AV20
GA90



Z-IMPR.
N-REGR.

PUCIC MAN

(pagina seguinte)

É um jogo idêntico ao "labirinto" e necessário fazer o percurso completo sem se deixar apanhar pelo seu perseguidor.

Tal como no jogo "aliens" (publicado neste número), os gráficos são definidos no fim do programa. Pode no entanto começar a passar o programa pela linha 2000, possibilitando-lhe a execução dos gráficos.

NOTA (GRÁFICOS)

Nas linhas:

- 72, 260, 455, 460, 470, 475, 490 e 610-graphics A

- 240, 530, 630 e

640-graphics E

- 1090-graphics U, I, O e P

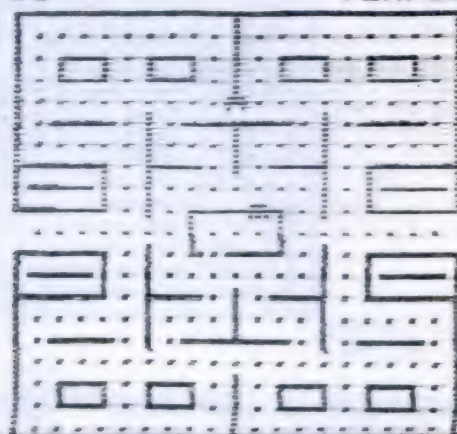
- 1500 a 1700

↑ - GRAPHICS A	↑ - GRAPHICS S
↑ - GRAPHICS D	↑ - GRAPHICS F
↑ - GRAPHICS G	↑ - GRAPHICS H
↑ - GRAPHICS J	↑ - GRAPHICS K
↑ - GRAPHICS L	

SCORE: 00

TEMPO3

=====



```

1 CLS
2 GO SUB 3000
10 GO TO 1000
20 LET P=P-(P<>0): BORDER P/3: LET A$=INKEY$
22 LET T=T+1: PRINT AT 0,26: INK 7:T
30 IF A$<>" " THEN LET D=(A$="7")+2*(A$="8")+3*(A$="6")+4*(A$="5")
40 IF D=0 THEN GO TO 110
50 LET A=ATTR(Y+B(D),X+A(D))
60 IF A=5 THEN GO TO 110
65 IF A=7 OR A=3 THEN LET S=S+10: LET K=K+1: BEEP ,007,0: PRINT AT 9,6: INK
70 IF K=194 THEN PRINT AT Y+B(D),X+A(D): INK 6:P$(D):AT Y,X: INK 0:" ": GO 3
= 300: GO TO 20
72 IF A=3 THEN LET P=20: PRINT AT YI,XI: INK 3: FLASH 1:" "
75 PRINT AT Y,X: INK 0:" "
80 LET X=X+A(D): LET Y=Y+B(D)
85 IF X=5 AND Y=11 AND D=4 THEN LET X=26
90 IF X=27 AND Y=11 AND D=2 THEN LET X=6
100 PRINT AT Y,X: INK 6:P$(D)
102 IF A=2 THEN GO TO 300
104 IF A=160 THEN GO TO 400
110 LET YY=Y-YI: LET XX=X-XI: IF P THEN LET XX=-XX: LET YY=-YY
111 IF D<>0 AND P=0 THEN LET YY=YY+B(D): LET XX=XX+A(D)
113 BEEP ,007,-2
115 IF ABS YY>ABS XX THEN GO TO 150
120 LET B=2+2*(XX<0)
130 LET C=1+2*(YY>0)
140 GO TO 170
150 LET B=1+2*(YY>0)
160 LET C=2+2*(XX<0)
170 FOR I=1 TO 4
175 LET A=ATTR(YI+B(I),XI+A(I))
180 IF A=6 AND P=0 THEN LET M=I: GO TO 240
182 IF A=6 AND P THEN GO TO 20
185 LET C(I)=(A=5)
190 NEXT I
200 IF C(D)=0 THEN LET M=B: GO TO 240
210 IF C(C)=0 THEN LET M=C: GO TO 240
220 IF C(B+2-4*(B>2))=0 THEN LET M=B+2-4*(B>2): GO TO 240
230 IF C(C+2-4*(C>2))=0 THEN LET M=C+2-4*(C>2): GO TO 240
235 GO TO 20
240 PRINT AT YI,XI: INK 7*(II=7)+3*(II=3):" "
245 IF P THEN BEEP ,007,-P
250 LET XI=XI+A(M): LET YI=YI+B(M)
255 LET II=ATTR(YI,XI): IF II=6 AND P=0 THEN GO TO 300
257 IF II=6 AND P THEN GO TO 400
260 PRINT AT YI,XI: INK 2: FLASH (P<>0):" "
270 GO TO 20
300 FOR I=1 TO 4
310 PRINT AT Y,X: INK 6:P$(I)

```



```

320 BEEP .1,I*5+2*Y
330 NEXT I: PRINT AT Y,X;" "
340 LET Y=Y-1
350 IF Y>0 THEN GO TO 300
360 BEEP 2,-5
370 CLS: PRINT AT 9,6; INVERSE 0;"PERDEU ESTE JOGO";AT 11,6;"O SEU SCORE ";S
375 PRINT AT 21,0;"QUALQUER TECLA PARA CONTINUAR": PAUSE 0: GO TO 10
400 LET P=0
410 BORDER 7: LET S=S+100+10*(II=7)
420 PRINT AT 0,6;S
430 BEEP 1,-5
440 BORDER 0
445 IF XI=16 THEN GO TO 465
450 FOR I=XI TO 16 STEP SGN (16-XI)
455 LET A=ATTR (YI,I): PRINT AT YI,I; INK 2; OVER 1;" "
460 BEEP .1,I: PRINT AT YI,I; INK A; OVER 1;" ": NEXT I
465 IF YI=11 THEN GO TO 480
467 FOR I=YI TO 11 STEP SGN (11-YI)
470 LET A=ATTR (I,16): PRINT AT I,16; OVER 1; INK 2;" "
475 BEEP .1,I: PRINT AT I,16; INK A; OVER 1;" ": NEXT I
480 LET XI=16: LET YI=11
485 LET II=ATTR (YI,XI)
490 PRINT AT YI,XI; INK 2;" "
495 GO TO 20
500 PRINT AT 0,26; INK 2; FLASH 1;T
502 LET S=S+60: IF T>500 THEN GO TO 504
504 PRINT AT 0,0;T;FLASH 1; INK 2;S
503 LET S=S+(500-T)
505 FOR A=2 TO 20
510 FOR B=6 TO 26
530 IF ATTR (A,B)<>5 THEN PRINT AT A,B; INK 7;" ": BEEP .005,10+A+B/21
540 NEXT B
550 NEXT A
560 PRINT AT 11,5; INK 0;" ";AT 11,26; INK 0;" "
570 LET II=7: LET K=0
580 LET X=16: LET Y=5: LET D=3
590 LET XI=16: LET YI=11
600 PRINT AT Y,X; INK 6;P*(D)
610 PRINT AT YI,XI; INK 2;" "
620 PRINT AT 0,6;S
625 LET T=0
627 PRINT AT 0,26;"0 "
630 PRINT AT 2,7; INK 3;" ";AT 2,25; INK 3;" "
640 PRINT AT 20,7; INK 3;" ";AT 20,25; INK 3;" "
650 LET P=0
650 RETURN
650 LET P=0
650 RETURN
665 POKE 23658,0
670 BORDER 0: CLS
680 DIM A(4): DIM B(4): DIM C(4)
690 DATA 0,-1,1,0,0,1,-1,0
700 RESTORE 1030
710 FOR I=1 TO 4
720 READ A(I),B(I)
730 NEXT I
740 LET S=0
750 LET P#=""
760 RESTORE 1500
770 PRINT #0;AT 1,11; FLASH 1; PAPER 6; INK 2;"ZX PUCKMAN"
780 FOR I=1 TO 21
790 READ A#
800 FOR J=1 TO LEN A#
810 PRINT AT I,J+5; INK 6;P*(2);AT I,J+5;

```



```

1160 PRINT INK 5*(ASC(J)-"0")/ASC(J)
1170 NEXT J: NEXT I
1180 PRINT AT 0,0: INK 7:"SCORE: 0"
1185 PRINT AT 0,21: INK 7:"TEMPO: 0"
1190 GO SUB 505: GO TO 20

```

```

1500 DATA "
1510 DATA "
1520 DATA "
1530 DATA "
1540 DATA "
1550 DATA "
1560 DATA "

```

```

1570 DATA "
1580 DATA "
1590 DATA "
1600 DATA "
1610 DATA "
1620 DATA "
1630 DATA "
1640 DATA "
1650 DATA "
1660 DATA "
1670 DATA "
1680 DATA "
1690 DATA "
1700 DATA "

```

```

2000 PRINT FLASH 1;TAB 7;"DESLIGUE O GRAVADOR";TAB 31
2005 RESTORE 2500
2010 FOR I=1 TO 16
2020 READ A$

```

```

2010 FOR I=1 TO 16
2020 READ A$

```

```

2030 FOR J=USR A$ TO USR A$+7
2040 READ A: POKE J,A
2050 NEXT J
2060 NEXT I
2070 RETURN

```

```

2500 DATA "A",24,24,24,31,31,24,24,24
2510 DATA "S",24,24,24,248,248,24,24,24
2520 DATA "D",0,0,0,255,255,24,24,24
2530 DATA "F",24,24,24,24,24,24,24,24
2540 DATA "G",0,0,0,255,255,0,0,0
2550 DATA "H",24,24,24,31,31,0,0,0
2560 DATA "J",0,0,0,31,31,24,24,24
2570 DATA "K",0,0,0,248,248,24,24,24
2580 DATA "L",24,24,24,248,248,0,0,0
2590 DATA "Q",0,60,126,90,255,219,219,219
2600 DATA "U",0,66,102,255,255,126,126,24
2610 DATA "I",24,126,124,248,248,124,126,24
2620 DATA "O",24,126,126,255,255,102,66,0
2630 DATA "P",24,126,62,31,31,62,126,24
2640 DATA "E",0,0,0,24,24,0,0,0
2650 DATA "M",24,24,24,255,255,0,0,0
2660 RETURN

```

```

1500 DATA "
1510 DATA "
1520 DATA "
1530 DATA "
1540 DATA "
1550 DATA "
1560 DATA "
1570 DATA "
1580 DATA "
1590 DATA "
1600 DATA "
1610 DATA "
1620 DATA "
1630 DATA "
1640 DATA "
1650 DATA "
1660 DATA "
1670 DATA "
1680 DATA "
1690 DATA "
1700 DATA "

```


SPECTRUM

USAP5, por Sinclair User
Out. 1982

Tens o controlo de uma
base com laser, que
usas para destruires
os invasores to
planeti.

Com as teclas 5 e 8
podes fazer movimentos
para a esquerda e para
a direita
respectivamente.

A tecla 7 e para
DISPARAR.

O numero de invasores
e limitado pelo grau
de dificuldade que
escolheres.

```

2 LET HS=0
3 POKE 23609,255
5 GO SUB 4000
10 GO SUB 1000
15 GO SUB 2000
20 LET C=15
25 LET X=0.0125
30 LET H=INT (RND*6)
40 LET I=INT (RND*20)
45 LET K=0
50 LET S=0
60 LET IN=INT (RND*7): IF IN>7
THEN LET IN=6
70 PRINT BRIGHT 1; INK 1; AT 19
,C;"N/A"; AT 18,C+1;"T"
75 PRINT AT 0,0; INK 7;"Score
";S
80 PRINT BRIGHT 1; INK IN; AT H
,I;"I"; BEEP .01,40
85 IF H=18 AND I=C+1 THEN GO TO
0 300
90 IF I=C AND H=19 THEN GO TO
300
95 IF H=19 AND I=C+1 THEN GO TO
0 300
97 IF H=19 AND I=C+2 THEN GO TO
0 300
100 IF INKEY$="7" THEN GO TO 20
0
110 IF INKEY$="8" THEN LET C=C+
(C<29): BEEP .01,40
120 IF INKEY$="5" THEN LET C=C-
(C>1): BEEP .01,30
130 PRINT AT 19,C-1;" "; AT
18,C-1;" "; AT H,I;" "; AT
135 LET H=H+1: IF H=20 THEN LET
K=K+1
140 IF H=20 THEN LET H=1: IF K=
2 THEN GO TO 3000
145 LET A=RND
150 IF A<.4 THEN LET I=I+2: IF
A>.5 THEN LET I=I-1
155 IF I>30 THEN LET I=1: IF I<
1 THEN LET I=25
160 GO TO 60
200 BEEP X,4: BEEP X,6: BEEP X,
8: BEEP X,10: BEEP X,12: BEEP X,
14: BEEP X,16: BEEP X,4: BEEP X,
5: BEEP X,8: LET A=C+1: LET E=17
210 PRINT INK 5; AT E,A;"B"
215 BEEP 0.01,5
220 IF E=H AND A=I THEN PRINT A
T E,A; INK 4;"U": BEEP .75,6: PA
USE 8: PRINT INK 3; AT E,A;"X":
LET S=S+5: PAUSE 5: PRINT AT H,C
;" "; AT E,A;" "; LET I=I+5 AND
I<25: LET H=INT (RND*10): GO TO
70
230 PRINT AT E,A;" "
240 LET E=E-1
250 IF E=1 THEN GO TO 100
260 GO TO 210
300 PRINT FLASH 1; INK 1; AT 19,
I-1;"X"; AT 18,I-1; INK 6;"X"
: BEEP 4,-15
305 PRINT AT 0,0;"
"
310 PRINT PAPER 1; INK 7; FLASH
1; AT 3,0;"O SEU LAZER FOI DESTR
UIDO"
320 PRINT INK 4; AT 4,0;"O SEU S
CORE ";S;" PONTOS"
321 IF S>HS THEN LET HS=S: INPU
T FLASH 1; PAPER 7; INK 1;"ULTRA
PASSOU O RECORD,
ESCRE
VA O SEU NOME ";N$
322 IF HS>0 THEN PRINT INK 6; AT
5,0;"RECORD: ";HS;" PONTOS - ";
N$: GO TO 400
325 IF S=0 THEN GO TO 370: GO TO
0 400
340 FOR S=1 TO 5/5
345 LET P=INT (RND*57): IF P>0 O
R P=0 THEN GO TO 345
350 PRINT INK P;"I";
355 NEXT P
360 PRINT INK 6;"ALIEN": PRINT
INK 6;"S" AND S/5
365 GO TO 400
370 PRINT INK 9; PAPER 7;"NAO A

```


GRAFICOS (Programa ALIENS)

ALIENS (Continuação)

```

400>INPUT INK 7;"NOVO JOGO ?";A
$
410 LET A$=A$+" "
420 IF A$(1)="S" OR A$="s" THEN
CLS : GO TO 15
430 STOP
1000 FOR F=0 TO 7
1010 READ A: POKE USR "I"+F,A
1020 NEXT F
1030 FOR F=0 TO 7
1040 READ A: POKE USR "B"+F,A
1050 NEXT F
1060 FOR F=0 TO 7
1070 READ A: POKE USR "U"+F,A
1080 NEXT F
1090 FOR F=0 TO 7
1100 READ A: POKE USR "N"+F,A
1110 NEXT F
1120 FOR F=0 TO 7
1130 READ A: POKE USR "A"+F,A
1140 NEXT F
1150 FOR F=0 TO 7
1160 READ A: POKE USR "T"+F,A
1170 NEXT F
1180 RETURN
1300 DATA 0,60,90,60,60,60,30,0
1400 DATA 0,0,15,55,15,15,15,0
1500 DATA 0,36,66,60,60,90,60,0
1600 DATA 1,3,7,15,29,63,90,90
1700 DATA 126,192,224,240,164,25
2,6,6
1800 DATA 8,0,0,0,0,24,60,126
2000 BORDER 0: PAPER 0: INK 7: C
LS : FOR F=0 TO 100
2010 LET A=INT (RND*225)
2020 LET B=INT (RND*150)
2030 IF A>255 OR B>195 THEN GO T
O 2010
2040 PLOT INK 7; BRIGHT 1;A,B: B
EEP .01,RND: NEXT F
2050 RETURN
3000 CLS : PRINT INK 3;AT 1,0;"D
EIXOU AVANÇAR MUITOS INIMIGOS. C
ONSEGUIRAM DOMINAR TERRA."
3010 GO TO 320
4000 BORDER 1: PAPER 1: CLS : PR
INT INK 6;TAB 11,"ALIENS"
4010 PRINT INK 7;AT 2,0;"DEVE DE
FENDER TERRA ABATENDO OS INIMIGO
S. PARA UTILIZAR O LAZER USE: TE
CLAS- 5 e 6          7- PARA
DISPARAR."
4020 PRINT INK 7;"OBTEN 5 PONTOS
POR CADA INIMIGO ABATIDO."
4030 PRINT INK 7;AT 10,0;"ESCOLH
A GRAU DE DIFICULDADE:";AT 12,0;
"F- FACIL";AT 13,0;"M- MEDIO";AT
14,0;"D- DIFICIL"
4040 INPUT A$
4050 GO TO (4060 AND A$="F" OR A
$="f")+ (4070 AND A$="M" OR A$="m
")+ (4080 AND A$="D" OR A$="d")
4060 GO TO 4040
4070 PRINT INK 3: BORDER 2: NAO E
CORRE O Jogo PASSAR MAIS QUE 10 IN
IMIGOS LET 2=20 PAUSE 2000 DE
TURN
4080 PRINT INK 1: PAPER 7: NAO E
CORRE O Jogo PASSAR MAIS QUE 5 INI
MIGOS LET 1=5 PAUSE 2000 DE
TURN
4090 PRINT INK 3: BORDER 2: NAO E
CORRE O Jogo PASSAR MAIS QUE 5 INI
MIGOS LET 1=5 PAUSE 2000 DE
TURN

```

No passar este
programa, não consegue
desenhar os graficos
das linhas 70, 80,

210, 220 e 350.

Isto porque eles são
definidos pelas
instruções DATA nas
linhas 1300-1800.

No entanto, nos
espaços reservados aos
graficos, use o

comando "graphics"

(caps shift e

graphics) e a tecla

referente ao grafico

que quer definir:

70 - graphics n, GR

Symbol shift 8, GR 9,

GR t

80 - GR i

210 - GR b

220 - GR n, GR Symbol

shift 6 (2 vezes)

300 - GR Symbol shift

6

350 - GR i

NOTA: GR significa
graphics

1000 1000 SPECTRUM

(In. YOUR COMPUTER.

Agosto/82)

Com este programa pode
fazer do seu SPECTRUM
um "mini orgão",
conseguindo uma escala
musical completa e no
tom desejado.

Logo no início do
programa tem 3 opções -
a, i e d - que lhe
permitem uma escala
mais grave ou mais
aguda. Em seguida pode
optar por um som
vibrado ou não (S/N).

Tem agora à sua frente,
no ecran, o esquema de
um teclado normal,
fazendo-se corresponder
uma nota a cada tecla.

Finalmente, se pegar no
nosso boletim anterior
(numero 8), encontrara
na pag. 20 uma forma de
obter o som mais alto
do seu SPECTRUM.

```

3 PRINT "Escolha um dos tres
tons: m, u ou d"
4 INPUT q$: IF q$="m" THEN LE
T k=0
5 IF q$="u" THEN LET k=12
6 IF q$="d" THEN LET k=-12
7 CLS
8 INPUT "quer o som vibrado ?
";v$
9 IF v$="s" OR v$="S" THEN GO
TO 11
10 LET x=0.3: GO TO 15
11 LET x=0.03
15 PRINT : PAPER 0; INK 6; AT 3
,4;"0"; AT 3,6;"U"; AT 3,8;"E"; AT
3,10;"R"; AT 3,12;"T"; AT 3,14;"Y"
; AT 3,16;"U"; AT 3,18;"I"; AT 3,20
;"O"; AT 3,22;"P"
17 PRINT AT 2,5;"@"; AT 2,7;"@";
; AT 2,9;"@"; AT 2,11;"@"; AT 2,1
5;"@"; AT 2,19;"@"; AT 2,21;"@"
18 PRINT AT 19,0;"USE AS TECLA
S INDICADAS"
19 PRINT AT 20,0;"Utilize as m
inusculas"; AT 21,0;"A- para term
inar"
20 IF INKEY$="a" THEN STOP
21 IF INKEY$="l" THEN BEEP x,0
+k
22 IF INKEY$="6" THEN BEEP x,1
+k
23 IF INKEY$="y" THEN BEEP x,2
+k
24 IF INKEY$="7" THEN BEEP x,3
+k
25 IF INKEY$="u" THEN BEEP x,4
+k
26 IF INKEY$="i" THEN BEEP x,5
+k
27 IF INKEY$="9" THEN BEEP x,6
+k
28 IF INKEY$="o" THEN BEEP x,7
+k
29 IF INKEY$="8" THEN BEEP x,8
+k
30 IF INKEY$="p" THEN BEEP x,9
+k
31 IF INKEY$="r" THEN BEEP x,-
1+k
32 IF INKEY$="4" THEN BEEP x,-
2+k
33 IF INKEY$="e" THEN BEEP x,-
3+k
34 IF INKEY$="3" THEN BEEP x,-
4+k
35 IF INKEY$="w" THEN BEEP x,-
5+k
36 IF INKEY$="2" THEN BEEP x,-
6+k
37 IF INKEY$="q" THEN BEEP x,-
7+k
40 GO TO 20

```


SPECTRUM

CLIMAS

GRAFICOS OMBROTERMICOS

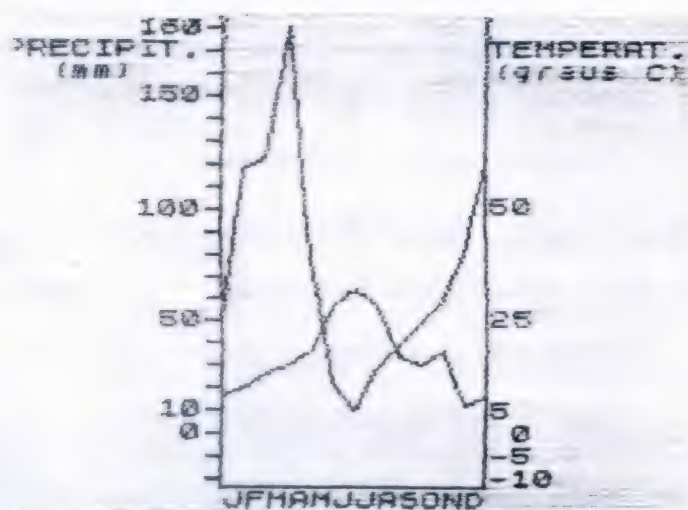
O programa "climas", essencialmente didactico, destina-se a desenhar (e ensinar a desenhar) graficos ombrotermicos, isto e, graficos duplos de temperatura e precipitacao.

Comeca por desenhar os eixos das coordenadas com a respectiva escala, indicada apenas com alguns valores numericos para nao sobrecarregar...

A escala nao atinge certos valores extremos que se podem verificar em um ou outro clima; e isso foi feito propositadamente para nao "comprimir" demasiadamente os graficos. Mas pode-se entrar com esses valores sem "enravar" o programa: isso obtem-se com as linhas 110 e 120. Para tais casos, no entanto, o grafico nao corresponderia aos valores numericos.

NOTA

E preciso entrar DUAS VEZES com os valores de temperatura e precipitacao referentes a Janeiro; o proprio programa pede isso - e so introduzir o que ele for pedindo!



```

10 FOR J=0 TO 20: PRINT TAB 9;
  "-": NEXT J
20 PRINT AT 1,0;"PRECIPIT.";AT
  1,22;"TEMPERAT.";AT 2,2;"(mm)";
  AT 2,22;"(graus C)";AT 21,10;"JF
  MAMJJASOND"
30 PRINT AT 0,6;"180";AT 3,6;"
  150";AT 6,6;"100";AT 8,22;"50";A
  T 13,7;"50";AT 13,22;"25";AT 17,
  7;"10";AT 17,22;"5";AT 18,8;"0";
  AT 18,23;"0";AT 19,22;"-5";AT 20
  ,22;"-10"
40 PLOT 79,175: DRAW 0,-167: D
  RAW 96,0: DRAW 0,167
50 DATA "Janeiro","Fevereiro",
  "Marco","Abril","Maio","Junho",
  "Julho","Agosto","Setembro","Outu
  bro","Novembro","Dezembro","Jane
  iro"
60 DIM x(13): DIM y(13)
70 FOR v=1 TO 2: RESTORE : FOR
  n=1 TO 13: READ M$
80 IF v=1 THEN INPUT "Temperat
  ura de ";(M$);"? ";t: LET y(n)=2
  7+t*.6
90 IF v=2 THEN INPUT "Precipit
  acao de ";(M$);"? ";p: LET y(n)=2
  7+p*.8
100 LET x(n)=71+n*.8
110 IF y(n)>175 THEN LET y(n)=1
  75
120 IF y(n)<0 THEN LET y(n)=0
130 IF n=1 THEN PLOT x(n),y(n)
140 IF n>1 THEN DRAW x(n)-x(n-1
  ),y(n)-y(n-1)
150 NEXT n: NEXT v


```


METODO DE PREVISAO segundo o amortecimento exponencial simples de BROWN

EXPLICACAO TEORICA

O METODO a seguir apresentado serve para estabelecer a previsao de um ou mais valores futuros de uma serie, com um minimo de erro, sendo para isso usados os proprios valores da serie com pesos decrescentes exponencialmente.

Assim a serie deve ser apresentada como uma funcao em que as abcissas sao os tempos (com um periodo bem definido) e as ordenadas sao os valores de alguns acontecimentos ao longo desse periodo.

 EQUACAO DO METODO
$$\hat{Z}_{t+T} = M_t = a \sum_{i=0}^{T-1} Z_{t-i} + (1-a)M_{t-1}$$

$$M_t = \frac{1}{N} \sum_{i=0}^{N-1} Z_{t-i}$$

No programa apresentado e por simplificacao optamos por considerar $T = 1$ pois usualmente e necessaria a obtencao de previsoes passo a passo.

Assim :

$$\hat{Z}_{t+1} = M_t = a \sum_{i=0}^{N-1} Z_{t-i} + (1-a)M_{t-1}$$

$$M_t = \frac{\sum_{i=0}^{N-1} Z_{t-i}}{N}$$

A primeira escolha a efectuar e a do ANO INICIAL DO CALCULO, ou seja o periodo a partir do qual vamos comecar a estabelecer previsoes com o objectivo de fazer a primeira previsao util, ou seja uma previsao de um periodo imediatamente superior ao ultimo da serie.

A segunda escolha sera o valor de (a) isto e o peso que daremos aos valores tomados para estabelecer uma previsao $0 < a < 1$

A terceira escolha sera a de N isto e o tamanho A2. Este valor da indicacao da quantidade de valores da serie que vamos usar, para fazer uma previsao.

Em cada previsao que e feita estabelece-se o valor do erro cometido na previsao atraves da expressao

$$E_t = \left| Z_t - \hat{Z}_t \right|$$

Assim ao longo de todas as previsoes feitas podemos ter todos os erros e calcular o erro quadratico medio atraves de

$$E_{q.m.} = \frac{\sum_{i=1}^P (E_i)^2}{PE - 1}$$



P TOTAL DE PERIODOS

PE TOTAL DE PREVISOES EFECTUADAS

O OBJECTIVO DESTE METODO E OBTER UMA PREVISAO O MAIS CORRECTA POSSIVEL, OU SEJA, COM O MENOR ERRO QUADRATICO MEDIO. Para que isso aconteca, vamos tentar minimiza-lo da seguinte forma

1) variar o 'ano inicial de calculo'.....desde o periodo $t=N+1$ ate $t=final$ da serie

EM CADA ANO INICIAL DE CALCULO

2) variar N de 1 a N

EM CADA ANO INICIAL DE CALCULO E POR CADA N

3) variar $\langle a \rangle$ em acrescimos consecutivos desde ... 0.1 ate 1

EXEMPLO

ANO	VALOR	
1958	44.4	Z1
1959	50.5	Z2
1960	51.6	Z3
1961	52.0	Z4
1962	55.3	Z5
1963	63.2	Z6
1964	72.1	Z7
1965	83.6	Z8
1966	97.8	Z9
1970	118.7	Z10
1971	134.4	Z11
1972	142.7	Z12

SEJA : $T=1$
 $N=4$
 INICIO EM 1962
 $a = 0.3$

valor previsto em Z13 = 114.7 para 1970 com um erro quadratico medio de

1052.45 A PARTIR DESTE VALOR QUE E GRANDE e que e obtido no computador, vamos fazer variar o $\langle a \rangle$, N, e ano inicial do calculo segundo a ordem atras mencionada, ate obter um, ERRO QUADRATICO MEDIO <MINIMO>.

PROGRAMAMETODO DE PREVISÃO

Computador NEW BRAIN

RUI PEIXO /A. JOUSA

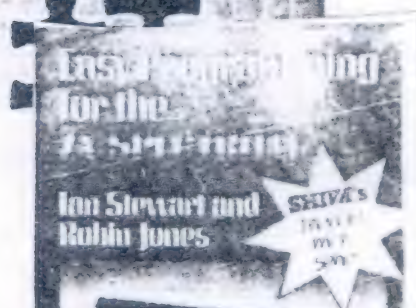
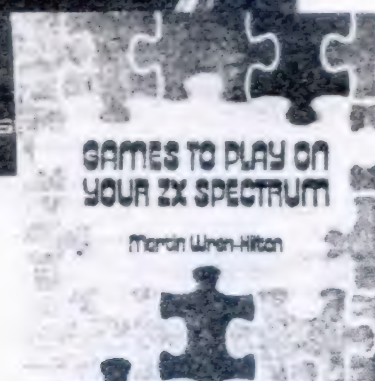
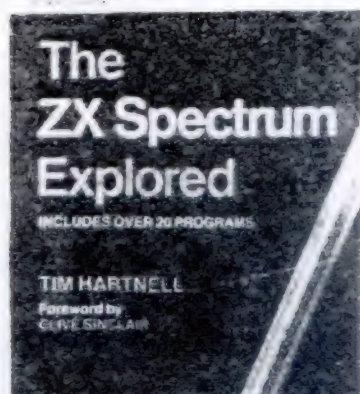
```

10 REM TABELA ANO/VALOR
12 DIMN(30),V(30),E(30)
15 I=0
20 PRINT"ANO = ";:INPUTN(I)
25 PRINT"VALOR = ";:INPUT V(I)
30 IFV(I)<0THEN GOTO 100
35 I=I+1
40 GOTO20
100 REM ATRIBUICAO VALOR ALFA/TAM.
101 M1=0
102 A3=I-1
105 PRINT"VALOR DE ALFA=": INPUTA1
110 PRINT"TAM.=": INPUTA2
200 REM CALCULO INICIAL
205 PRINT"O ANO INICIAL DO CALCULO DEVE COMECAR N+1 ANOS APÓS O INICIO DA TAB
ELA":INPUTN1
210 IFN1<N(A2+1)THEN PRINT"ANO SEGUINTE ...!":GOTO205
215 FORJ=1TOI-1
220 IF M1=N(J)THEN GOTO300
225 NEXTJ
230 PRINT"NAO PODEMOS CONSIDERAR ESTE ANO"
235 INPUTY$:GOTO200
300 FOR K=1TOA2
305 LETM1=M1 +V(J):J=J+1
310 NEXTK
320 LETM1=M1/A2
325 Z=A1*V(J)+A1*M1
326 J=J+1
330 L=0
340 GOSUB5000
2000 M1=Z
2010 Z=A1*V(J)+A1*M1
2017 J=J+1
2020 GOSUB5000
2026 IFJ=A3-1 THEN GOTO3000
2030 GOTO2000
3000 E2=0
3010 FORP=0 TO L-1
3020 E2=E2+E(P)^2
3030 NEXTP
3040 E2=E2/L
3042 PRINT"VALOR PREVISTO=";Z
3045 PRINT"ERRO QUADRATICO MEDIO=";E2
3060 PRINT"QUER NOVO CALCULO ? (S/N)": INPUTY$
3065 IFY$="S" THEN GOTO 100
4000 STOP
5000 REM CALCULO DO ERRO
5010 E1=V(J)-Z
5020 E(L)=E1:L=L+1
5030 RETURN
10002 END

```


20 BEST PROGRAMS

for the
ZX SPECTRUM
WITH EXPLANATORY TEXT



=====

LIVROS SPECTRUM

BREVE PANORAMICA

=====

GAMES TO PLAY ON YOUR ZX SPECTRUM

Martin Wren-Hilton
Shiva Publishing Ltd
43 paginas

38 programas, na sua maioria jogos, com algumas notas explicativas. São quase todos muito pequenos, não conseguindo assim demonstrar as características especiais do ZX Spectrum.

EASY PROGRAMMING FOR THE ZX SPECTRUM

Ian Stewart e Robin Jones
Shiva Publishing Ltd
139 paginas

Um conjunto de conhecimentos de BASIC são aqui transmitidos de um modo bastante encorajador para aqueles que pretendam avançar.

Alguns programas dedicam-se exclusivamente a ilustrar pontos específicos, apesar de serem relativamente pequenos.

Embora o conteúdo fundamental do livro gire a volta do BASIC, as características do ZX Spectrum não deixam de ser referidas.

FERNANDO A. PRECES

=====

No numero anterior fechamos o artigo no ponto em que iam os alterar novamente a rotina em pseudo-Basic. Aqui esta:

Rotina completa em Pseudo-Basic, ja incluindo a sub-rotina PRINT AT.

```
1002 LET DE = 16578
1004 LET STACK 1 = DE
1005 GOTO 1060
1010 LET DE = STACK 1
1015 LET A = PEEK DE
1016 LET CP1 = A-23
1017 IF NOT CP1 THEN GOTO 1045
1020 LET CP2 = A-216
1025 IF NOT CP2 THEN GOTO 1098
1030 PRINT CHR$ A;
1035 LET DE = DE+1
1040 GOTO 1015
1050 LET STACK 1 = DE
1055 GOTO 1065
1060 LET HL = 16514
1065 LET B = PEEK HL
1070 LET HL = HL+1
1075 LET C = PEEK HL
1080 LET HL = HL+1
1085 LET STACK 2 = HL
1090 PRINT AT B,C;
1095 GOTO 1010
1098 STOP
```

Nota: Tem sido uma preocupacao em programas ja apresentados fazer-se uma pequena analise teorica, sempre que surgem instrucoes que possam suscitar duvidas ao leitor. Apesar de algumas dessas instrucoes ja terem sido abordadas por outros artigos nesta revista do CLUBE Z-80, atendendo a que so muito recentemente entraram para o CLUBE uma grande parte dos seus membros, pensa o autor que deve, com risco de se tornar repetitivo, fazer um pequeno resumo teorico sobre a aplicacao dessas instrucoes. Assim, a seguir vamos falar um pouco sobre o (STACK), a pilha.

Existe na memoria dos computadores uma certa area denominada (STACK) PILHA. Esta area e usada como espaco de trabalho, para armazenamento de

THE ZX SPECTRUM EXPLORED

Tim Hartnell
Sinclair Browne Ltd
218 paginas

Sao utilizadas cerca de 46 listagens para demonstrar as caracteristicas do ZX Spectrum em BASIC.

As listagens sao facsimiles da impressora Sinclair, dirigidas a applicacoes comerciais, educacionais e ludicas. Nenhuma se pode considerar sensacional; no entanto constituem material razoavel e suficiente para acompanhar um programador inexperiente. O conteudo parece realmente justificar o titulo do livro.

OVER THE SPECTRUM

Philip Williams
Melbourne House Ltd
164 paginas

Uma compilacao de 28 programas em BASIC, alguns baseados tambem em codigo maquina, sempre acompanhados das necessarias explicacoes.

A unica critica que deve fazer-se e que alguns dos programas sao tao engenhosos que se torna dificil detectar erros efectuados ao passá-los. Ha algumas implementacoes de graficos do ZX Spectrum muito bem conseguidas.

Mais do que uma fonte de programas para copiar, OVER THE SPECTRUM e um livro para o fazer pensar.

endereços ou dados, que somente são utilizados numa base temporária. Cada nova entrada é empilhada sobre a anterior numa forma muito especial.

O Z80 usa uma pilha que cresce na memória no sentido descendente, ou seja, vai formando a pilha empurrando com a nova entrada, a anterior cada vez mais para o topo da memória.

O (STACK POINTER) anotador de pilha é um dos registos do microprocessador Z80 que é utilizado pela unidade de control para apontar as diferentes localizações da pilha, decrescendo uma unidade de localização (contagem inversa) por cada nova entrada introduzida na pilha.

Quando um dado é retirado da pilha, o anotador é incrementado de uma unidade. Todos os movimentos de dados que envolvam a pilha requerem um manejo de 2 bytes separados. Um dado permanece quieto na pilha até que é empurrado pela chegada de um outro.

O STACK é muito usado quando se trabalha com programação em código máquina. Das muitas instruções existentes para trabalhar com a pilha, duas delas são muito importantes:

As primeiras são para empurrar 2 bytes de dados para o STACK. As segundas são para trazer estes dois bytes de dados do STACK.

Quando uma operação PUSH é criada, o STACK POINTER (apontador de pilha) registo SP é primeiro decrementado, e uma cópia do alto byte é armazenada na localização apontada por ele, voltando a ser decrementado pela segunda vez, para armazenar na localização imediata o baixo byte.

Uma acção inversa dá-se durante uma operação POP.

É importante saber que quando 2 bytes são empurrados para o STACK não fica qualquer

informação sobre a sua origem, pelo que uma instrução POP os traz de volta para dentro de qualquer registo, a escolha do programador.

Estas instruções são utilizadas sempre que os registos em causa tenham de trabalhar outros dados, para não se perder a informação neles contida, que vai ficar armazenada no STACK.

Esta acção é levada a cabo nesta rotina em Pseudo-Basic; o programa em código máquina vai necessitar de usar a ROM (Rotinas PRINT CHR\$ e PRINT AT) e, portanto, precisa de preservar as informações que nesse momento estão contidas nos registos DE e HL, visto esses registos irem ser utilizados pelas rotinas referidas.

Vamos agora ensaiar a nossa rotina, mas antes é necessário introduzir os parâmetros para a ROTINA DA ROM (PRINT AT).

Escreva:

```
1 REM 1234567890
```

Pelos processos já seus conhecidos transforme esta REM em 4 (números 1, 2, 3 e 4). Chame a ROTINA 1 e transforme estas REMs numa única (reserva de 59 caracteres para introduzir os parâmetros de PRINT AT e o código máquina).

Chame a Rotina 3 e introduza:

```
(2, 4, 6, 2, 9, 2, 11, 2, 13, 2, 15, 2, 256).
```

Experimente agora GOTO 1000...

Se tudo estiver certo vai obter os seguintes resultados:

```
linha 2, coluna 4; JOGO (ATAQUE EM TERRA)
```

```
linha 6, coluna 2; PONTUACAO
```

```
linha 9, coluna 2;
```

```
ASTEROIDES....10 PONTOS
```

```
Etc., etc.
```

A actuação do programa em Pseudo-Basic é agora um pouco mais rápida que as anteriores, mas jamais se aproximara da velocidade do programa em código máquina, que vamos introduzir em seguida.

Chame de novo a rotina 3 e escreva em código decimal.

16514	-	2	} Estes são os parâmetros para a rotina PRINT AT do RUN
15	-	4	
16	-	6	
17	-	2	
18	-	10	
19	-	2	
20	-	12	
21	-	2	
22	-	14	
23	-	2	
24	-	16	} LD DE, NN
25	-	2	
26	-	17	
27	-	194	
28	-	64	
29	-	213	
30	-	24	
31	-	18	
32	-	209	
33	-	26	
34	-	254	} LET DE = 16578 PUSH DE - LET STACK 1 = DE JR, e (18) } GOTO 1060 POP DE - LET DE = STACK 1 LDA, (DE) - LET A = PEEK DE CP, N (*) - LET CP1 = A-23 JZ, e (8) } IF NOT CP1 THEN GOTO 1045 CP, N } LET CP2 = A-216 (**) } RET Z - IF NOT CP2 THEN GOTO 1098 (0)
35	-	23	
36	-	40	
37	-	8	
38	-	254	
39	-	216	
40	-	200	
41	-	8	
42	-	215	
43	-	19	
44	-	24	} INC DE JR, e } GOTO 1015 (256-243)
45	-	243	
46	-	19	
47	-	213	
48	-	24	
49	-	3	
50	-	33	
51	-	130	
52	-	64	
53	-	70	
54	-	35	} LD B, (HL) - LET B = PEEK HL INC HL - LET HL = HL+1 LD C, (HL) - LET C = PEEK HL INC HL - LET HL = HL+1 PUSH HL - LET STACK2 = HL CALL, NN 229: } GOSUB 229: 233: } PRINT AT ROM 235: } 237: } 239: } 241: } 243: } 245: } 247: } 249: } 251: } 253: } 255: } 257: } 259: } 261: } 263: } 265: } 267: } 269: } 271: } 273: } 275: } 277: } 279: } 281: } 283: } 285: } 287: } 289: } 291: } 293: } 295: } 297: } 299: } 301: } 303: } 305: } 307: } 309: } 311: } 313: } 315: } 317: } 319: } 321: } 323: } 325: } 327: } 329: } 331: } 333: } 335: } 337: } 339: } 341: } 343: } 345: } 347: } 349: } 351: } 353: } 355: } 357: } 359: } 361: } 363: } 365: } 367: } 369: } 371: } 373: } 375: } 377: } 379: } 381: } 383: } 385: } 387: } 389: } 391: } 393: } 395: } 397: } 399: } 401: } 403: } 405: } 407: } 409: } 411: } 413: } 415: } 417: } 419: } 421: } 423: } 425: } 427: } 429: } 431: } 433: } 435: } 437: } 439: } 441: } 443: } 445: } 447: } 449: } 451: } 453: } 455: } 457: } 459: } 461: } 463: } 465: } 467: } 469: } 471: } 473: } 475: } 477: } 479: } 481: } 483: } 485: } 487: } 489: } 491: } 493: } 495: } 497: } 499: } 501: } 503: } 505: } 507: } 509: } 511: } 513: } 515: } 517: } 519: } 521: } 523: } 525: } 527: } 529: } 531: } 533: } 535: } 537: } 539: } 541: } 543: } 545: } 547: } 549: } 551: } 553: } 555: } 557: } 559: } 561: } 563: } 565: } 567: } 569: } 571: } 573: } 575: } 577: } 579: } 581: } 583: } 585: } 587: } 589: } 591: } 593: } 595: } 597: } 599: } 601: } 603: } 605: } 607: } 609: } 611: } 613: } 615: } 617: } 619: } 621: } 623: } 625: } 627: } 629: } 631: } 633: } 635: } 637: } 639: } 641: } 643: } 645: } 647: } 649: } 651: } 653: } 655: } 657: } 659: } 661: } 663: } 665: } 667: } 669: } 671: } 673: } 675: } 677: } 679: } 681: } 683: } 685: } 687: } 689: } 691: } 693: } 695: } 697: } 699: } 701: } 703: } 705: } 707: } 709: } 711: } 713: } 715: } 717: } 719: } 721: } 723: } 725: } 727: } 729: } 731: } 733: } 735: } 737: } 739: } 741: } 743: } 745: } 747: } 749: } 751: } 753: } 755: } 757: } 759: } 761: } 763: } 765: } 767: } 769: } 771: } 773: } 775: } 777: } 779: } 781: } 783: } 785: } 787: } 789: } 791: } 793: } 795: } 797: } 799: } 801: } 803: } 805: } 807: } 809: } 811: } 813: } 815: } 817: } 819: } 821: } 823: } 825: } 827: } 829: } 831: } 833: } 835: } 837: } 839: } 841: } 843: } 845: } 847: } 849: } 851: } 853: } 855: } 857: } 859: } 861: } 863: } 865: } 867: } 869: } 871: } 873: } 875: } 877: } 879: } 881: } 883: } 885: } 887: } 889: } 891: } 893: } 895: } 897: } 899: } 901: } 903: } 905: } 907: } 909: } 911: } 913: } 915: } 917: } 919: } 921: } 923: } 925: } 927: } 929: } 931: } 933: } 935: } 937: } 939: } 941: } 943: } 945: } 947: } 949: } 951: } 953: } 955: } 957: } 959: } 961: } 963: } 965: } 967: } 969: } 971: } 973: } 975: } 977: } 979: } 981: } 983: } 985: } 987: } 989: } 991: } 993: } 995: } 997: } 999: }

Após escrever
500 RAND USR 16526

o grave o programa

Faca GOTO 500. Deve obter no ecran uma imagem semelhante a obtida quando utilizou a rotina em pseudo-basic, mas agora com a alta velocidade do codigo maquina. Para verificar as diferencas de ocupacao de memoria entre as duas rotinas, se quiser, pode escrever:

```
600 INPUT X (a localizacao
      que pretender)
610 FOR N = X TO X + 500
620 LET B = PEEK N
630 PRINT N, CHR$ B
640 NEXT N
650 STOP
```

Esta pequena rotina mostra-lhe a localizacao pretendida e o seu conteudo. Sempre que o ecran encha e o relatorio 5 apareca, pode continuar a observacao premindo CONT.

Para o programa monitor ficar mais versatil, ate porque muitas rotinas em codigo maquina vem numa base numerica hexadecimal, pode introduzir no programa uma rotina que faz a entrada do Assembler nessa base.

Escreva:

```
125 PRINT, "4 - CODIGO MAQUINA EM HEXADECIMAL"
9600 PRINT, "INTRODUZA O CODIGO EM HEXADECIMAL"
9610 LET X = 16514
9620 PRINT, "PODE INTRODUIR EM SIMULTANEO VARIAS INSTRUcoes"
9625 LET A$ = " "
9630 IF A$ = " " THEN INPUT A$
9640 IF A$ = "S" THEN RETURN
9650 LET N = 16*CODE A$(1) + CODE A$(2) - 476
9660 PRINT AT 18, 2; X, N
9665 POKE X, N
9670 SCROLL
9675 LET X = X+1
9680 LET A$ = A$ (3 TO)
9685 GOTO 9630
```

Para pequenas conversoes entre numeros decimais e hexadecimais, tem a seguir uma tabela que lhe permitira conversoes ate ao decimal 255 - hexadecimal FF.

TABELA DE CONVERSAO DE NUMEROS
HEXADECIMAIS \longleftrightarrow DECIMAIS

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
2	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
3	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
4	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
5	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95
6	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111
7	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127
8	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143
9	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159
A	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175
B	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191
C	192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207
D	208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223
E	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239
F	240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255

(CONTINUA NO PROXIMO NUMERO)

EXISTE UMA FOLHA EM SEPARADO QUE DEVES PREENCHER E RESPONDER COM A MAIOR BREVIDADE.....SE POSSIVEL ONTEM!

ESTE INQUÉRITO DESTINA-SE A AJUDAR OS COORDENADORES DO NOSSO JORNAL NO SENTIDO DE SE CONSEGUIR PUBLICAR TEXTOS E PROGRAMAS COM INTERESSE.

DEVOLVE A FOLHA ANEXA AO CLUBE Z80, PARA QUE O JORNAL PASSE A CONTER PAGINAS COM INTERESSE DIRECTO PARA AQUELES QUE ESTAO AGRUPADOS NO CLUBE Z80.

A resposta devera especificar quais os microcomputadores com classificacao mais importante para o associado; igualmente sobre o tipo de programas, devera incidir uma classificacao

POR EXEMPLO :

- 1 - JOGOS COM LABIRINTOS
- 2 - PROGRAMAS SOBRE FICHEIROS
- 3 - " SOBRE MATEMATICA
- 4 - JOGOS DO TIPO AVENTURAS

No caso da classificacao dos textos, teremos casos como por exemplo :

- 1 - INICIACAO A PROGRAMACAO
- 2 - LINGUAGENS DE PROGRAMACAO
- 3 - APRECIACAO SOBRE PROGRAMAS DO SPECTRUM
- 4 - O COMPUTADOR E A EDUCACAO

SAO EXEMPLOS QUE VOS PODEM SERVIR DE GUIA. INSISTIMOS TODAVIA EM CONHECER A V/RESPOSTA PARA QUE O JORNAL POSSA EVOLUIR.

EM RELACAO AS OBSERVACOES, PODEREMOS POR EXEMPLO CONHECER O V/INTERESSE EM CONTINUAR LIGADO OU NAO AO CLUBE Z80, OU SUGESTOES DE INICIATIVAS NOVAS QUE DEEM VIDA A ESTA EXISTENCIA.

UM ABRACO DO CLUBE Z80

ATENÇÃO ESTAS RESPOSTAS SÃO APENAS EXEMPLOS



ANEXO (INQUÉRITO)

CLUBE 280
AV. BOAVISTA 832-2 T
4100 PORTO

INQUÉRITO
=====

QUE PROGRAMAS ??? QUE MAQUINAS ??? QUE TEXTOS ???

NOME

MAQUINA.....

PREFIRO PROGRAMAS PARA O MICRO TIPO : 1

2

3

4

OS PROGRAMAS COM MAIS INTERESSE :

1

2

3

4

TIPO DE TEXTOS QUE GOSTARIA DE VER PUBLICADOS :

1

2

3

4

OBSERVAÇÕES

.....
.....

as instruções encontram-se na página 28 do JORNAL DO CLUBE
280